Available online at www.sciencedirect.com

ScienceDirect

Discrete Mathematics 309 (2009) 5284–5297

DISCRETE MATHEMATICS

www.elsevier.com/locate/disc

Gray codes for column-convex polyominoes and a new class of distributive lattices

Stirling Chow, Frank Ruskey*

Department of Computer Science, University of Victoria, Canada

Received 20 July 2006; accepted 9 November 2007
Available online 7 January 2008

Abstract

We introduce the problem of polyomino Gray codes, which is the listing of all members of certain classes of polyominoes such that successive polyominoes differ by some well-defined closeness condition (e.g., the movement of one cell). We discuss various closeness conditions and provide several Gray codes for the class of column-convex polyominoes with a fixed number of cells in each column. For one of our closeness conditions, a natural new class of distributive lattice arises: the partial order is defined on the set of m-tuples $[S_1] \times [S_2] \times \cdots \times [S_m]$, where each $S_i > 1$ and $[S_i] = \{0, 1, \ldots, S_i - 1\}$, and the cover relations are $(p_1, p_2, \ldots, p_m) \prec (p_1 + 1, p_2, \ldots, p_m)$ and $(p_1, p_2, \ldots, p_j, p_{j+1}, \ldots, p_m) \prec (p_1, p_2, \ldots, p_j - 1, p_{j+1} + 1, \ldots, p_m)$. We also discuss some properties of this lattice.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Polyomino; Gray code; Hamilton cycle; Distributive lattice

1. Introduction

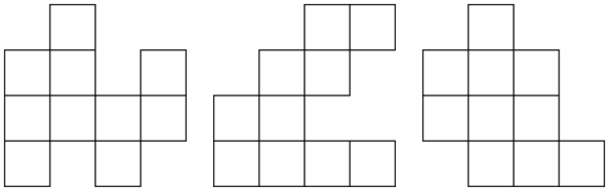
A combinatorial Gray code is a complete and non-repeating listing of the members of some class of combinatorial objects such that successive objects in the listing differ by a well-defined closeness condition. For example, the classic Binary Reflected Gray Code is a listing of all 2^n binary strings of length n such that successive strings differ by a single bit flip.

A *polyomino* is an edge-connected set of unit squares, called *cells*, embedded in the plane. We will assume that the embedding is such that the edges of the squares are aligned with the x- or y-axis. Polyominoes are often classified by area and referred to as n-ominoes when they contain n cells. For example, the games of dominoes and Tetris are played with 2-ominoes and 4-ominoes (tetrominoes), respectively.

Polyominoes have been extensively studied and have a wide range of applications in mathematics and the physical sciences [5,7,4]. The problem of counting n-ominoes has garnered considerable interest [9,2,3], and although counts up to 47-ominoes are known (see sequence $A001168$ [11]), the problem remains open.

* Corresponding author. Tel.: +1 250 721 7232.
E-mail address: ruskey@cs.uvic.ca (F. Ruskey).
URL: http://www.cs.uvic.ca/~ruskey (F. Ruskey).

0012-365X/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.disc.2007.11.038



(a) Column-convex not row-convex. (b) Row-convex not column-convex. (c) Convex.

Fig. 1. 10-ominoes that exhibit different convexity properties.

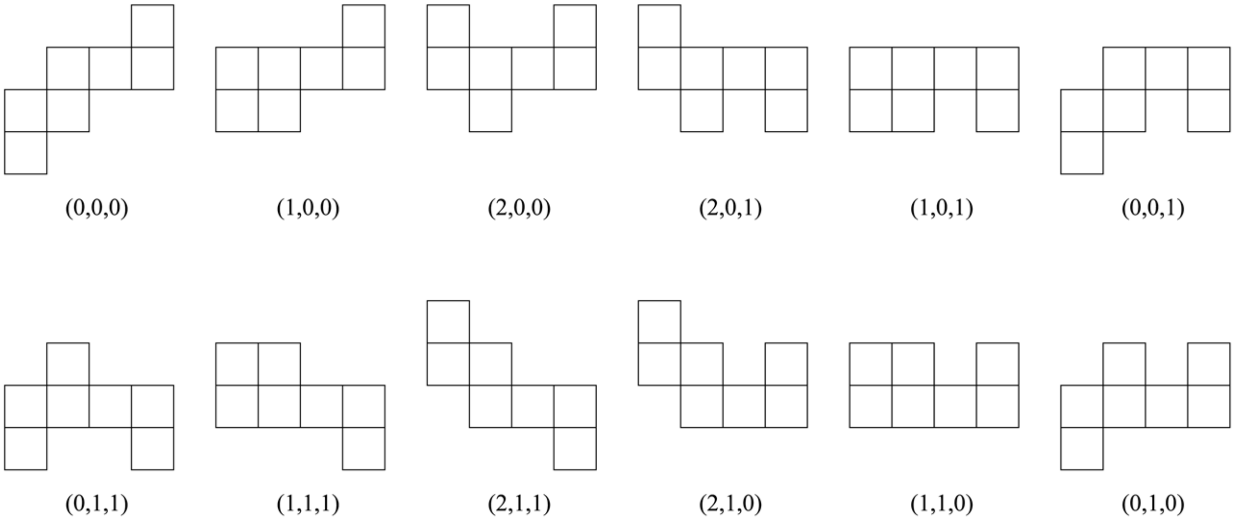


Fig. 2. All [2, 2, 1, 2]-ominoes and their offset representations.

Many subclasses of polyominoes have been defined. If every intersection of a polyomino with a vertical line is connected, then the polyomino is called *column-convex*. If every intersection of a polyomino with a horizontal line is connected, then the polyomino is called *row-convex*. A *convex* polyomino is one that is both column-convex and row-convex (see Fig. 1).

Closed-form expressions for the size of most classes of polyominoes are unknown, which suggests that the problem of finding Gray codes for these classes is difficult. Our initial step towards discovering polyomino Gray codes is to consider classes whose enumeration is relatively easy.

In this paper, we do not distinguish polyominoes that are translations of each other. In other words, a translation of a polyomino does not create a new polyomino, but a rotation or reflection can.

Let $\mathbf{P}_n$ be the set of all column-convex n-ominoes and consider any polyomino $\mathbf{p} \in \mathbf{P}_n$. When ordered from left to right, the number of cells in the columns of $\mathbf{p}$ induces an integer composition of n with all parts positive. For example, the columns of Fig. 1(a) induce the composition $10 = 3 + 3 + 2 + 2$.

Definition 1.1. A column-convex n-omino that induces the composition $n = a_1 + a_2 + \cdots + a_{k+1}$, where $a_j \geq 1$ for all $j = 1, 2, \ldots, k + 1$, is referred to as an $[a_1, a_2, \ldots, a_{k+1}]$-omino. We use $\mathbf{a}$ to denote $[a_1, a_2, \ldots, a_{k+1}]$.

For example, Fig. 2 shows the set of all [2, 2, 1, 2]-ominoes. The reason for using $k + 1$ instead of k in the above definition will become clear later.

Theorem 1.2. *The number of* $[a_1, a_2, \ldots, a_{k+1}]$*-ominoes is* $\prod_{j=1}^{k}(a_j + a_{j+1} - 1)$*, where the empty product is* 1.

Proof. We induct on k. For $k = 0$, any $[a_1]$-omino is a single column of a_1 cells; there is only one such polyomino. For the inductive step, consider adding a column of a_{k+1} cells to the right of each of the $\prod_{j=1}^{k-1}(a_j + a_{j+1} - 1)$ different $[a_1, a_2, \ldots, a_k]$-ominoes. There are $a_k + a_{k+1} - 1$ ways to do this that preserve the edge-connectedness property

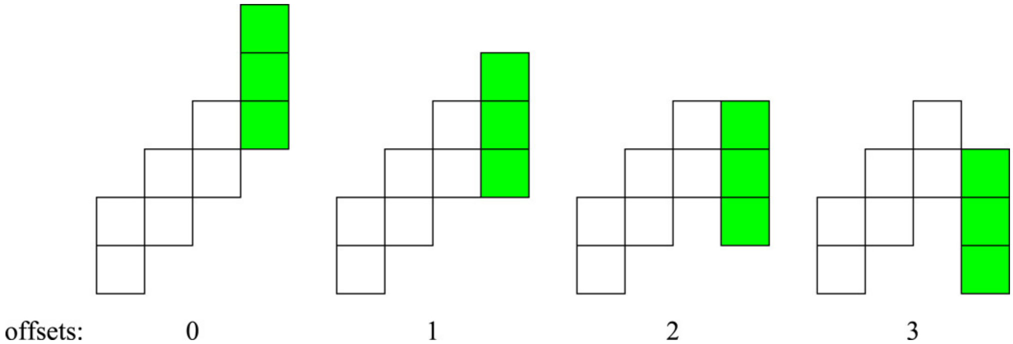


Fig. 3. The four ways to add a 3-cell column to the end of a [2, 2, 2]-omino.

(see Fig. 3), and so there are

$$(a_k + a_{k+1} - 1)\prod_{j=1}^{k-1}(a_j + a_{j+1} - 1) = \prod_{j=1}^{k}(a_j + a_{j+1} - 1)$$

$[a_1, a_2, \ldots, a_{k+1}]$-ominoes. □

In the remainder of this paper, we only consider **a**-ominoes, and not any wider class of polyominoes.

2. Representing a-ominoes

In a Gray code for the **a**-ominoes, each polyomino induces the same composition, so it is the vertical orientation of the columns that differentiates successive polyominoes. By using the relative orientation of each column with respect to the column to its left, we can derive a representation for the **a**-ominoes that is position-independent and shape-centric.

Definition 2.1. Let **p** be an $[a_1, a_2, \ldots, a_{k+1}]$-omino and define the *offset* of any two adjacent columns j and $j+1$ to be an integer whose value is ordered from 0, when column $j+1$ is oriented as high above column j as possible, to $a_j + a_{j+1} - 2$, when column $j+1$ is oriented as far below column j as possible. The *offset representation* of **p** is the tuple $(p_1, p_2, \ldots, p_k)$, where p_j is the offset of columns j and $j+1$. We identify **p** and $(p_1, p_2, \ldots, p_k)$, and use them interchangeably.

Fig. 3 shows the $2 + 3 - 1 = 4$ possible offsets of two columns with 2 and 3 cells, respectively, and the label under each polyomino in Fig. 2 gives that polyomino's offset representation.

Definition 2.2. Let $\langle S_1, S_2, \ldots, S_k\rangle$ be the product space $[S_1] \times [S_2] \times \cdots [S_k]$, where $S_j \geq 1$ and $[S_j] = \{0, 1, \ldots, S_j - 1\}$ for all $j = 1, 2, \ldots, k$. We use **S** to denote the product space $\langle S_1, S_2, \ldots, S_k\rangle$. We may also consider **S** as a k-tuple of positive integers, either implicitly (when the context is clear), or explicitly with the term *sequence* **S**.

For the $[a_1, a_2, \ldots, a_{k+1}]$-ominoes, the offset representation is a bijection between the polyominoes and the tuples of the product space $\langle S_1, S_2, \ldots, S_k\rangle$ where

$$S_j = a_j + a_{j+1} - 1. \tag{1}$$

We define a map ϑ that takes $(k+1)$-tuples of positive integers to k-tuples of positive integers by $\vartheta([a_1, a_2, \ldots, a_{k+1}]) = \langle S_1, S_2, \ldots, S_k\rangle$, where S_j satisfies Eq. (1) for all $j = 1, 2, \ldots k$. For example, the offset representation establishes a bijection between the [2, 2, 1, 2]-ominoes and the product space $\vartheta([2, 2, 1, 2]) = \langle 3, 2, 2\rangle$.

Using the offset representation, the problem of generating a Gray code for the **a**-ominoes is equivalent to listing the tuples of the product space $\vartheta(\mathbf{a})$ such that consecutive tuples map to polyominoes that differ by the prescribed closeness condition. It is important to note that two different sets of column composition polyominoes can map to

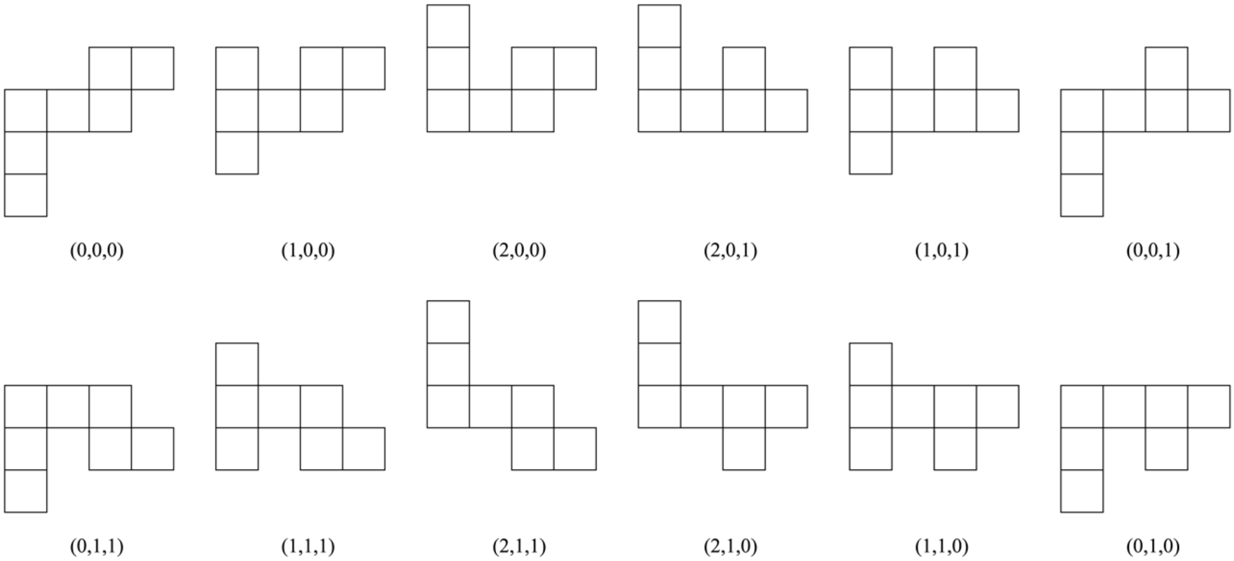


Fig. 4. All [3, 1, 2, 1]-ominoes; these map to the same product space, $\langle 3, 2, 2\rangle$, as the [2, 2, 1, 2]-ominoes in Fig. 2.

the same product space. For example, the [2, 2, 1, 2]-ominoes and [3, 1, 2, 1]-ominoes both map to the product space $\langle 3, 2, 2\rangle$ as illustrated by Figs. 2 and 4.

Furthermore, there are product spaces $\mathbf{S}$ for which $\vartheta^{-1}(\mathbf{S})$ is empty. For example, consider $\langle S_1, S_2, S_3\rangle = \langle 1, 2, 1\rangle$, and suppose there exist positive integers a_1, a_2, a_3, a_4 such that $\vartheta([a_1, a_2, a_3, a_4]) = \langle 1, 2, 1\rangle$. Since $S_1 = 1$ implies that $a_1 = a_2 = 1$, and $S_3 = 1$ implies that $a_3 = a_4 = 1$, we have $S_2 = a_2 + a_3 - 1 = 1$, which contradicts $S_2 = 2$; therefore, $\vartheta^{-1}(\langle 1, 2, 1\rangle)$ is empty.

Definition 2.3. The product space $\mathbf{S} = \langle S_1, S_2, \ldots, S_k\rangle$ is *feasible* if $\vartheta^{-1}(\mathbf{S})$ is non-empty (i.e., the set of Eqs. (1) for all $j = 1, 2, \ldots, k$ has a solution in positive integers $a_1, a_2, \ldots, a_{k+1}$).

A simple substitution process will reveal whether or not a product space is feasible. In the remainder of this paper, we only consider Gray codes for **a**-ominoes, and we use a general strategy based on listing the tuples of the corresponding product space $\vartheta(\mathbf{a})$.

3. Gray codes of a-ominoes

We refer to the transition between successive objects in a Gray code as a *move*. When we define a Gray code, we must describe what operations can comprise a move and what closeness condition the move must adhere to. In addition, there might be global conditions that the Gray code must satisfy. For example, in a binary Gray code, the operation might be a bit flip, the closeness condition might be no more than two bit flips between successive binary strings, and the global condition might be that the first string must be all 0's and the last string must be all 1's. This section describes some of the Gray code properties that apply to **a**-ominoes; however, in many cases these can be generalized to other classes of polyominoes.

Definition 3.1. We define the following two types of *offset operations* on k-tuples of integers:

- $\tau_j^{\pm}((p_1, p_2, \ldots, p_k)) = (p_1, \ldots, p_j \pm 1, \ldots, p_k),\ 1 \le j \le k$
- $\sigma_j^{\pm}((p_1, p_2, \ldots, p_k)) = (p_1, \ldots, p_j \mp 1, p_{j+1} \pm 1, \ldots, p_k),\ 1 \le j < k.$

If the k-tuple $(p_1, p_2, \ldots, p_k)$ corresponds to a polyomino $\mathbf{p}$, then the τ_j operation can be thought of as cutting $\mathbf{p}$ between columns j and $j + 1$ and shifting the two pieces up or down one cell relative to each other. Shifting a contiguous block of m columns up one cell can also be interpreted as moving m cells from the bottom of the block (one from each column) to the top, or vice versa for moving the block down.

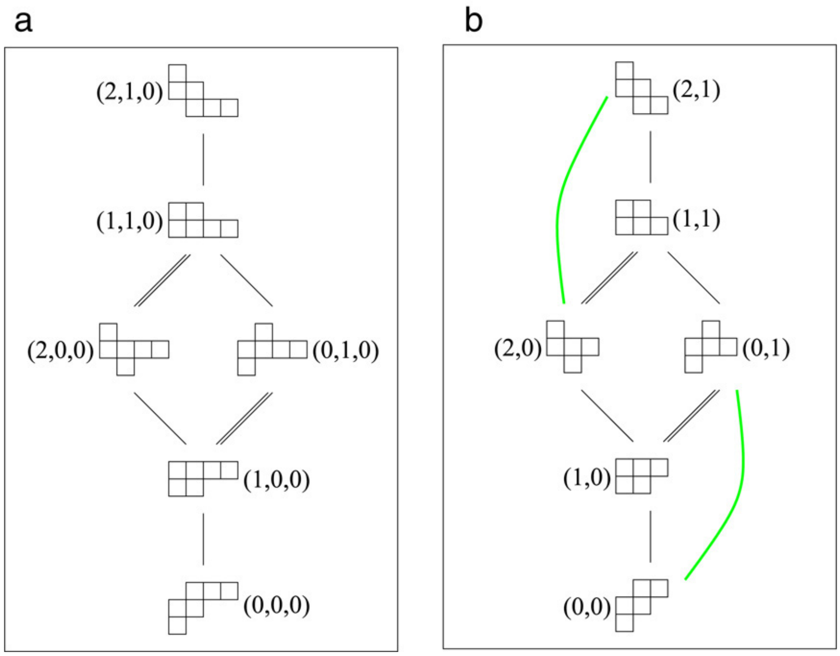


Fig. 5. The graphs $G([2, 2, 1, 1]) \cong G(\langle 3, 2, 1\rangle)$ and $G([2, 2, 1]) \cong G(\langle 3, 2\rangle)$.

The σ_j operation can be thought of as cutting out column $j + 1$ and shifting it up or down one cell relative to the rest of $\mathbf{p}$; alternatively, it can be thought of as moving one cell from the top of column $j + 1$ to the bottom of column $j + 1$, or vice versa. Of course, the σ operation can be thought of as the composition of two τ operations: $\sigma_j^+(\mathbf{p}) = \tau_j^-(\tau_{j+1}^+(\mathbf{p}))$.

This section presents some results concerning the existence of certain **a**-omino Gray codes. Graph-theoretic properties of the underlying Gray code "closeness" graph are also discussed.

Since there is a bijection between certain product spaces and $[a_1, a_2, \dots, a_{k+1}]$-ominoes, it is tempting to use existing Gray codes for product spaces to infer Gray codes for such polyominoes. Gray codes for product spaces are described in Knuth [6] and Williamson [1]; while these Gray codes use the τ operation to obtain successive tuples in the code, the corresponding polyominoes can differ by the movement of a large number of cells. In particular, a τ_j move causes either j cells to move or $k - j$ cells to move; since this is not a constant number of cell moves, we reject it as a closeness condition for polyominoes. We desire Gray codes where a constant number of cells (preferably only one), move between successive polyominoes.

The σ operations can be thought of as causing only one cell to move; however, we cannot use only σ operations because they leave the sum $p_1 + p_2 + \cdots + p_k$ invariant, and thus cannot generate the entire product space. We therefore augment the σ operations with the two "extreme" τ operations: τ_1 and τ_k. These extreme τ operations need to move only one cell.

Define an undirected graph $G(\mathbf{a})$ whose vertices are the **a**-ominoes and whose edges join polyominoes that differ by the movement of one cell. A Hamilton path in $G(\mathbf{a})$ corresponds to an exhaustive sequencing of the **a**-ominoes such that successive polyominoes differ by the movement of one cell.

Also, define an undirected graph $G(\mathbf{S})$ whose vertices are the k-tuples of $\mathbf{S}$ and whose edges join tuples that differ by the application of an operation from the set $\{\tau_1, \tau_k, \sigma_1, \sigma_2, \dots, \sigma_{k-1}\}$. By our previous discussion, if $\vartheta(\mathbf{a}) = \mathbf{S}$, then $G(\mathbf{a}) \cong G(\mathbf{S})$, where $\cong$ denotes graph isomorphism. We think of the edges of $G(\mathbf{S})$ as being labelled by the (unsigned) operation that takes one of the incident vertices to the other incident vertex.

Fig. 5 shows $G([2, 2, 1, 1])$ and $G([2, 2, 1])$, where the vertices are labelled with the associated offset representations. Edges drawn with a single line are τ edges, and those drawn with a double line are σ edges. For the remainder of this section, we deal with product spaces $\mathbf{S}$, irrespective of whether or not they are feasible. Let $\mathbf{S}^R$ denote the reversal of the sequence $\mathbf{S}$. For example, $\langle 3, 2, 1\rangle^R = \langle 1, 2, 3\rangle$. The proofs of the following three lemmas are trivial and are omitted.

Lemma 3.2. *For any product space* $\mathbf{S}$,

$$G(\mathbf{S}^R) \cong G(\mathbf{S}).$$

If $S_j = 1$, then $G(\mathbf{S})$ has no τ_j, σ_{j-1}, or σ_j edge; this leads to the next two lemmas.

Lemma 3.3. *If* $S_{j-1} = S_j = 1$, *then*

$$G(\langle S_1, S_2, \ldots, S_{j-1}, S_j, S_{j+1}, \ldots, S_{k-1}, S_k\rangle) \cong G(\langle S_1, S_2, \ldots, S_{j-1}, S_{j+1}, \ldots, S_{k-1}, S_k\rangle).$$

We can therefore assume that there are no successive 1's in $\mathbf{S}$.

Lemma 3.4. *If* $S_j = 1$, $1 < j < k$, *then*

$$G(\langle S_1, S_2, \ldots, S_{k-1}, S_k\rangle) \cong G(\langle S_1, S_2, \ldots, S_{j-1}, 1\rangle) \times G(\langle 1, S_{j+1}, \ldots, S_{k-1}, S_k\rangle),$$

where $\times$ *denotes the cartesian product of the two graphs.*

The previous lemma inspires us to define $\mathbf{S}$ to be *prime* if $S_j = 1$ implies that $j = 1$ or $j = k$. If $\mathbf{S}$ is prime, then $G(\mathbf{S})$ is also said to be prime. By Lemma 3.4, any $G(\mathbf{S})$ can be factored into a cartesian product of prime graphs. Let us say that a prime $\mathbf{S}$ is *left-anchored* if $S_1 = 1$ and *right-anchored* if $S_k = 1$. If $\mathbf{S}$ is both left- and right-anchored, then we say that it is *frozen*. If $\mathbf{S}$ is neither left- nor right-anchored, then we say that it is *free*. Note that the terms left-anchored, right-anchored, frozen, and free each imply primality.

We now consider the question of when $G(\mathbf{S})$ is connected. For example, $G([1, 1, n, 1, 1]) \cong G(\langle 1, n, n, 1\rangle)$ has $2n - 1$ connected components. The "rank" function, defined in (2) below, will be later shown to give the distance from $(0, 0, \ldots, 0)$ to $\mathbf{p} = (p_1, p_2, \ldots, p_k)$ in $G(\mathbf{S})$, where $\mathbf{S}$ is right-anchored and not frozen.

$$r(\mathbf{p}) := \sum_{j=1}^{k} j p_j. \tag{2}$$

Observe that

$$r(\tau_1^+(\mathbf{p})) = r(\sigma_j^+(\mathbf{p})) = 1 + r(\mathbf{p}). \tag{3}$$

Lemma 3.5. *The graph* $G(\mathbf{S})$ *is connected if and only if* $\mathbf{S} = \langle S_1, S_2, \ldots, S_k\rangle$ *has no frozen prime factors or* $k = 1$.

Proof. Note that, by Lemma 3.3, we assume that $\mathbf{S}$ has no successive 1's.

If $\mathbf{S}$ has a frozen prime factor $\mathbf{F}$ and $k \neq 1$, then $\mathbf{F} = \langle 1, \mathbf{S}', 1\rangle$, where $\mathbf{S}'$ is a subsequence of $\mathbf{S}$ which does not contain any 1's. In addition, $\mathbf{S}'$ is non-empty since, otherwise, $\mathbf{F} = \langle 1, 1\rangle$, which contradicts our assumption that $\mathbf{S}$ has no successive 1's. $\mathbf{0} = (0, 0, \ldots, 0)$ is an isolated vertex in $G(\mathbf{F})$, and since $\mathbf{S}'$ is a non-empty sequence of integers greater than one, $G(\mathbf{F})$ has more than one vertex, and thus is disconnected. By Lemma 3.4, $G(\mathbf{S})$ is the cartesian product of prime graphs, one of which will be the disconnected graph $G(\mathbf{F})$. Since the cartesian product of any graph with a disconnected graph is also disconnected, $G(\mathbf{S})$ is disconnected.

If $k = 1$, then, trivially, $G(\mathbf{S})$ connected. If $k > 1$ and $\mathbf{S}$ has no frozen prime factors, then either $\mathbf{S}$ is prime or $\mathbf{S}$ has two factors where the left factor is right-anchored and the right factor is left-anchored. We first deal with the case where $\mathbf{S}$ is prime. By Lemma 3.2, we can assume that $\mathbf{S}$ is not left-anchored. Let $\mathbf{p} = (p_1, p_2, \ldots, p_k) \in \mathbf{S}$. Note that $\mathbf{0}$ is the only tuple in $\mathbf{S}$ for which $r(\mathbf{p}) = 0$. We will show that there is a path in $G(\mathbf{S})$ from $\mathbf{p}$ to $\mathbf{0}$ that decreases the value of r by one at each step. Let j be the smallest index such that $p_j > 0$; if there is no such j, then $\mathbf{p} = \mathbf{0}$. If $j = 1$, then reduce p_1 by one by applying the τ_1^- operation to $\mathbf{p}$. If $j > 1$, then reduce p_j by one and set p_{j-1} to 1 by applying the σ_{j-1}^- operation to $\mathbf{p}$. Thus $G(\mathbf{S})$ is connected since any two vertices are connected to $\mathbf{0}$ by a path.

If $\mathbf{S}$ has two factors where the left factor is right-anchored and the right factor is left-anchored, then by Lemma 3.4, $G(\mathbf{S})$ is the cartesian product of two graphs. By the discussion in the preceding paragraph, the product is that of the two connected graphs. Since the product of two connected graphs is also connected, $G(\mathbf{S})$ is connected. □

Implicit in the proof of the previous lemma is the following corollary.

Corollary 3.6. *If* $\mathbf{S}$ *is right-anchored and not frozen, then the distance in* $G(\mathbf{S})$ *from* $\mathbf{0}$ *to* $\mathbf{p}$ *is* $r(\mathbf{p})$.

Proof. If $\mathbf{S}$ is right-anchored and not frozen, then the edges of $G(\mathbf{S})$ are labelled by τ_1 or σ_j, $1 \le j < k$, (but not τ_k). Thus, by (3), there can be no shorter path than the one used in the proof of Lemma 3.5. □

Lemma 3.7. *The graph* $G(\mathbf{S})$ *is bipartite if* k *is odd, or if* $S_1 = 1$*, or if* $S_k = 1$.

Proof. Let $\mathbf{p} \in \mathbf{S}$ and consider the parity of $r(\mathbf{p})$:

$$r(\mathbf{p}) \equiv \sum_{j \text{ odd}} p_j \pmod 2.$$

Clearly, the operations τ_1 and σ_j change the parity of the rank (i.e., mod 2, $r(\mathbf{p}) \equiv 1 + r(\tau_1(\mathbf{p})) \equiv 1 + r(\sigma_j(\mathbf{p}))$). The operation τ_k will change the parity if k is odd; otherwise, it will leave it unchanged.

Thus, if k is odd, the parity of $r(\mathbf{p})$ determines the partite sets that show $G(\mathbf{S})$ to be bipartite. Similarly, if $S_k = 1$, then the τ_k operation cannot be applied, and so there are no τ_k edges in $G(\mathbf{S})$, and thus it is bipartite. If $S_1 = 1$, then we can apply the same argument to the graph $G(\langle S_k, \ldots, S_2, S_1\rangle) = G(\mathbf{S}^R) \cong G(\mathbf{S})$. □

Lemma 3.8. *If* $k > 1$ *is even, then* $G(\mathbf{S})$ *is not bipartite when all* $S_j > 1$ *(i.e., when* $\mathbf{S}$ *is free).*

Proof. Consider the path that starts at $\mathbf{0}$ and successively follows the edges labelled

$$\tau_1,\ \sigma_1,\ \ldots,\ \sigma_{k-1},\ \tau_k.$$

The vertices along the path are

$$00\cdots00,\ 10\cdots00,\ 01\cdots00,\ \ldots,\ 00\cdots10,\ 00\cdots01,\ 00\cdots00,$$

so it is, in fact, a cycle, and since each $S_j > 1$, the cycle exists in $G(\mathbf{S})$. Lastly, since k is even, the cycle has odd length, so $G(\mathbf{S})$ is not bipartite. □

Note that the two previous lemmas give us necessary and sufficient conditions for $G(\mathbf{S})$ to be bipartite, if $\mathbf{S}$ is prime.

We will think of $\mathbf{S}$ as a signed set according to the parity of $r(\mathbf{p})$. Define the *parity difference* $\mathrm{d}(\mathbf{S})$ of $\mathbf{S}$ to be

$$\mathrm{d}(\mathbf{S}) := |\{\mathbf{p} \in \mathbf{S} : r(\mathbf{p}) \text{ is even }\}| - |\{\mathbf{p} \in \mathbf{S} : r(\mathbf{p}) \text{ is odd}\}|.$$

Theorem 3.9.

$$\mathrm{d}(\mathbf{S}) = \begin{cases} 0 & \text{if some } S_j \text{ is even, where } j \text{ is odd,} \\ \prod_{j \text{ even}} S_j & \text{otherwise.} \end{cases}$$

Proof. First, suppose that some S_j is even, where j is odd. Define a sign-reversing involution ψ on $\mathbf{S}$ as follows: change p_j to $p_j + 1$ if p_j is even, and change p_j to $p_j - 1$ if p_j is odd. Clearly, $\psi(\psi(\mathbf{p})) = \mathbf{p}$ and ψ is sign-reversing. Furthermore, ψ has no fixed points, and thus $\mathrm{d}(\mathbf{S}) = 0$.

Otherwise, we may assume that S_j is odd for all odd j. Define another sign-reversing involution ϕ as follows. Let i be the smallest odd index such that $p_i \neq 0$. Let $\phi(\mathbf{p}) = \mathbf{q}$ be the same as $\mathbf{p}$ in every position except i. The value of q_i is $p_i + 1$ if p_i is odd and is $p_i - 1$ if p_i is even. Since the range of p_i is $0 < p_i < S_i$ and S_i is odd, q_i will satisfy $0 \le q_i < S_i$. Clearly, ϕ satisfies $\phi(\phi(\mathbf{p})) = \mathbf{p}$ and is sign-reversing. The fixed points of this involution occur when all the odd-indexed p_i are 0. Each of these will be in the even set, and therefore the parity difference is equal to the number of fixed points, namely,

$$\mathrm{d}(\mathbf{S}) = \prod_{j \text{ even}} S_j. \quad \square$$

Corollary 3.10. *If* k *is odd, or if* $S_1 = 1$*, or if* $S_k = 1$*, then* $G(\mathbf{S})$ *has no Hamilton path if* S_{2i+1} *is odd for all* i*, unless* $S_2 = S_3 = \cdots = S_{k-1} = 1$.

Proof. By Lemma 3.7, $G(\mathbf{S})$ is bipartite, with partite sets determined by the parity of the rank. Thus, there will be no Hamilton path if the product in Theorem 3.9 is greater than 1. This product is 1 only if all the even-indexed S_j are equal to 1. By Lemma 3.5, $G(\mathbf{S})$ is disconnected unless $S_2 = S_3 = \cdots = S_{k-1} = 1$. In this case, $G(\mathbf{S})$ is a two-dimensional grid graph of size $S_1 \times S_k$, and grid graphs always have Hamilton paths. $\square$

In what follows, we will use S' to denote $S_j - 1$ because it occurs so often. The following lemma generalizes Proposition 1 of [10].

Lemma 3.11. *There is no Hamilton path in $G(\langle S_1, S_2, \ldots, S_k, 1\rangle)$ if*

$$r((S_1', S_2', \ldots, S_k')) = \sum_{j=1}^{k} jS' \tag{4}$$

has the same parity as $S_1 S_2 \cdots S_k$.

Proof. Note that $(0, 0, \ldots, 0)$ and $(S',S',\ldots,S',0)$ are pendant vertices in the bipartite graph $G(\langle S_1, S_2, \ldots, S_k, 1\rangle)$, and thus a Hamilton path must start and end at those vertices. By Corollary 3.6, the distance between those vertices is given by (4). Since $S_1 S_2 \cdots S_k$ is the number of vertices in $G(\langle S_1, S_2, \ldots, S_k, 1\rangle)$, the negation of the stated parity condition must hold if there is a Hamilton path. $\square$

So far, we have established some necessary conditions for the existence of Hamilton paths and cycles in $G(\mathbf{S})$. Sufficient conditions will have to wait for the distributive lattice discussion in the next section, but we can state one previously known result now.

Lemma 3.12. *The graph $G(\langle 2, \ldots, 2, 1\rangle)$, where there are n 2's, has a Hamilton path if and only if $\binom{n+1}{2}$ is odd and $n \neq 5$.*

Proof. This follows from the results of Savage, Shields, and West [10]. $\square$

4. Right-anchored polyominoes and a distributive lattice

In this section, we consider product spaces $\mathbf{S}$ that are right-anchored, and not frozen (i.e., $\mathbf{S} = \langle S_1, S_2, \ldots, S_{k-1}, 1\rangle$, where $k > 1$ and $S_j > 1$ for all $j = 1, 2, \ldots, k-1$). We will use m to denote $k-1$ because it occurs so often. The allowable operations are τ_1 and $\sigma_1, \ldots, \sigma_{m-1}$, and without loss of generality, we can consider m-tuples only (by dropping the final 1), if we restrict ourselves to these operations. Consider the following ordering on m-tuples in $\langle S_1, S_2, \ldots, S_m\rangle$:

$$(p_1, p_2, \ldots, p_m) \preccurlyeq (q_1, q_2, \ldots, q_m) \quad \text{iff} \quad \sum_{i=j}^{m} p_i \le \sum_{i=j}^{m} q_i \quad \text{for all } j = 1, 2, \ldots, m.$$

The ordering is clearly reflexive, antisymmetric and transitive, and thus defines a partial order. The poset is denoted $P(\langle S_1, S_2, \ldots, S_m\rangle)$. Note that we could also view this poset as an interval in a similarly-defined infinite poset on $\mathbb{N} \times \mathbb{N} \times \cdots$. It will prove convenient to have a notation for the partial sums that occur in the previous definition. We define

$$\vec{p}_j := \sum_{i=j}^{m} p_i.$$

Theorem 4.1. *The poset $P(\langle S_1, S_2, \ldots, S_m\rangle)$ is a distributive lattice.*

Proof. We define the meet and join operations. The join of $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ and $\mathbf{q} = (q_1, q_2, \ldots, q_m)$ is $\mathbf{p} \wedge \mathbf{q} = \mathbf{r} = (r_1, r_2, \ldots, r_m)$ where

$$r_j = \max(\vec{p}_j, \vec{q}_j) - \vec{r}_{j+1},$$

with $\vec{r}_{m+1}$ defined to be zero. With this definition, we have $\vec{r}_j = r_j + \vec{r}_{j+1} = \max(\vec{p}_j, \vec{q}_j)$, and thus $\mathbf{p} \preccurlyeq \mathbf{r}$ and $\mathbf{q} \preccurlyeq \mathbf{r}$. Furthermore, no values smaller than r_j could be used without violating our ordering. We do need to verify that $0 \le r_j \le S'_j$. Note that $r_j = \max(\vec{p}_j, \vec{q}_j) - \max(\vec{p}_{j+1}, \vec{q}_{j+1})$. Since $\vec{p}_j \ge \vec{p}_{j+1}$ and $\vec{q}_j \ge \vec{q}_{j+1}$, we have $\max(\vec{p}_j, \vec{q}_j) \ge \max(\vec{p}_{j+1}, \vec{q}_{j+1})$, and so $r_j \ge 0$. In the other direction, since $\vec{p}_j \le S'_j + \vec{p}_{j+1}$ and $\vec{q}_j \le S'_j + \vec{q}_{j+1}$, we have $\max(\vec{p}_j, \vec{q}_j) \le S'_j + \max(\vec{p}_{j+1}, \vec{q}_{j+1})$, and so $r_j \le S'_j$. The meet is defined similarly, except that minima rather than maxima are used.

Recall that min's and max's distribute:

$$\max(a, \min(b, c)) = \min(\max(a, b), \max(a, c)).$$

Note that

$$(y \wedge z)_j = \max(\vec{y}_j, \vec{z}_j) - \max(\vec{y}_{j+1}, \vec{z}_{j+1}),$$

and thus

$$\overrightarrow{(y \wedge z)}_j = \sum_{i \ge j} (y \wedge z)_i = \sum_{i \ge j} (\max(\vec{y}_i, \vec{z}_i) - \max(\vec{y}_{i+1}, \vec{z}_{i+1})) = \max(\vec{y}_j, \vec{z}_j).$$

Therefore

$$\begin{aligned}(x \vee (y \wedge z))_j &= \min(\vec{x}_j, \overrightarrow{(y \wedge z)}_j) - \min(\vec{x}_{j+1}, \overrightarrow{(y \wedge z)}_{j+1}) \\ &= \min(\vec{x}_j, \max(\vec{y}_j, \vec{z}_j)) - \min(\vec{x}_{j+1}, \max(\vec{y}_{j+1}, \vec{z}_{j+1})) \\ &= \max(\min(\vec{x}_j, \vec{y}_j), \min(\vec{x}_j, \vec{z}_j)) - \max(\min(\vec{x}_{j+1}, \vec{y}_{j+1}), \min(\vec{x}_{j+1}, \vec{z}_{j+1})) \\ &= \max(\overrightarrow{(x \vee y)}_j, \overrightarrow{(x \vee z)}_j) \\ &= ((x \vee y) \wedge (x \vee z))_j. \quad \square\end{aligned}$$

Theorem 4.2. *The cover relations of $P(\langle S_1, S_2, \ldots, S_m \rangle)$ are*

$$(p_1, p_2, \ldots, p_m) \prec \tau_1^+((p_1, p_2, \ldots, p_m)) \quad \textit{if } p_1 < S'_1 \quad \textit{and}$$
$$(p_1, p_2, \ldots, p_m) \prec \sigma_j^+((p_1, p_2, \ldots, p_m)) \quad \textit{if } p_j > 0 \textit{ and } p_{j+1} < S'_{j+1}.$$

Proof. If $\mathbf{p} = (p_1, p_2, \ldots, p_m)$, then the sequences $\mathbf{q}$ specified in the statement of the theorem as $\mathbf{p} \prec \mathbf{q}$ are precisely the cases where $\vec{\mathbf{q}}$ differs from $\vec{\mathbf{p}}$ in exactly one position, and in that position, the value in $\mathbf{q}$ is one greater. Thus, no poset element can lie properly between $\mathbf{p}$ and $\tau_1^+(\mathbf{p})$ or between $\mathbf{p}$ and $\sigma_j^+(\mathbf{p})$.

We now show that there are no other cover relations. If $\mathbf{p} \prec \mathbf{q}$, then there is a value j for which $\vec{\mathbf{p}}_i = \vec{\mathbf{q}}_i$ for $i > j$, and $\vec{\mathbf{p}}_j < \vec{\mathbf{q}}_j$. Let $\ell \le m$ be the largest index such that $p_\ell \ne 0$. If there is no such index ℓ, then note that $\mathbf{p} \prec \tau_1^+(\mathbf{p}) \preccurlyeq \mathbf{q}$. If there is such an ℓ, then note that $\mathbf{p} \prec \sigma_\ell^+(\mathbf{p}) \preccurlyeq \mathbf{q}$. $\square$

The implicit condition that $S_j > 1$ for all $j = 1, 2, \ldots, m$ is necessary in the previous theorem. For example, in the poset $P(\langle 3, 1, 3 \rangle)$, the element $(0, 0, 2)$ covers $(1, 0, 1)$.

Theorem 4.3. *The cover graph of $P(\langle S_1, S_2, \ldots, S_m \rangle)$ is isomorphic to $G(\langle S_1, S_2, \ldots, S_m, 1 \rangle)$.*

Proof. This follows immediately from Theorem 4.2. $\square$

The following useful theorem was proven by Pruesse and Ruskey [8] in the more general context of the basic words of an antimatroid. Here, we state the theorem (along with the proof since it is short and not well known), in the more restricted context of posets. The prism $G \times e$ of a graph G is obtained by taking two copies of G and adding a perfect matching between corresponding vertices in the two copies. We prefix each vertex in $G \times e$ with a plus (+) or minus (−) to indicate which copy of G is being referred to. If $\mathcal{P}$ is a poset and x is a minimal element of $\mathcal{P}$, then $\mathcal{P}/x$ denotes the poset with the element x (and all relations involving x) removed. The poset $\mathcal{P} \setminus x$ is the poset $\mathcal{P}$ with x and all elements $y \succeq x$ removed. Let I be an ideal of $\mathcal{P}$. Note that either I is an ideal of $\mathcal{P} \setminus x$, or $I \setminus \{x\}$ is an ideal of $\mathcal{P}/x$, according to whether or not I contains x.

Theorem 4.4 (*Pruesse–Ruskey*). *The prism of the cover graph of any distributive lattice is Hamiltonian.*

Proof. Let D be a distributive lattice. By Birkhoff's Theorem (e.g., as presented in [12], pg. 106), there is a poset $\mathcal{P}$ such that D is isomorphic to $J(\mathcal{P})$, the lattice of ideals of $\mathcal{P}$. Our proof is by induction on $|\mathcal{P}|$, the number of elements of $\mathcal{P}$. Let $G(\mathcal{P})$ denote the undirected cover graph of $J(\mathcal{P})$. We strengthen the inductive assumption to state that for every minimal element $x \in \mathcal{P}$, there is a Hamilton cycle in $G(\mathcal{P}) \times e$ that includes the edges $(-\emptyset, +\emptyset)$ and $(+\emptyset, +\{x\})$.

If $|\mathcal{P}| = 1$, then $G(\mathcal{P}) \times e$ is the 4-cycle $+\emptyset, +\{x\}, -\{x\}, -\emptyset$; otherwise, assume that $|\mathcal{P}| > 1$, and let x be minimal.

Inductively, there is a Hamilton cycle

$$+\emptyset = X_1, X_2, \dots, X_p = -\emptyset$$

in $G(\mathcal{P}/x) \times e$.

If x is the minimum, then the Hamilton cycle

$$+\emptyset, X_1 \cup \{x\}, X_2 \cup \{x\}, \dots, X_p \cup \{x\}, -\emptyset$$

satisfies the conditions of the theorem.

Otherwise, let x and y be minimal elements of $\mathcal{P}$. There are Hamilton cycles

$$+\emptyset = X_1, X_2, \dots, X_p = -\emptyset \text{ in } G(\mathcal{P}/x) \times e \quad \text{and}$$
$$+\emptyset = Y_1, Y_2, \dots, Y_q = -\emptyset \text{ in } G(\mathcal{P} \setminus x) \times e,$$

with $X_2 = Y_2 = +\{y\}$. The Hamilton cycle

$$\begin{aligned}+\emptyset = Y_1, X_1 \cup \{x\}, X_p \cup \{x\}, X_{p-1} \cup \{x\}, \dots, X_2 \cup \{x\},\\ Y_2, Y_3, \dots, Y_{q-1}, Y_q = -\emptyset\end{aligned}$$

satisfies the conditions of the theorem. $\square$

Theorem 4.5. *The graph $G(\langle S_1, S_2, \dots, S_m, x\rangle)$ is Hamiltonian if x is even.*

Proof. First, observe that $G(\langle S_1, \dots, S_m, 1\rangle) \times e$ is isomorphic to a spanning subgraph of $G(\langle S_1, S_2, \dots, S_m, 2\rangle)$, where the σ_m edges are missing and the τ_{m+1} edges correspond to prism edges. By Theorem 4.4, there is a Hamilton cycle in $G(\langle S_1, \dots, S_m, 1\rangle) \times e$, and thus there is a corresponding Hamilton cycle H in $G(\langle S_1, S_2, \dots, S_m, 2\rangle)$.

Let $\Upsilon = \{\tau_1, \sigma_1, \dots, \sigma_{m-1}\}$. The edges of $G(\langle S_1, \dots, S_m, 1\rangle) \times e$ are either Υ or prism edges. Since the prism edges form a perfect matching, there are no successive τ_{m+1} edges in H, and so every vertex in H is incident with an Υ edge. Hence, H contains edges e_0 and e_1 where

$$e_0 = ((q_1, q_2, \dots, q_m, 0), \kappa_0(q_1, q_2, \dots, q_m, 0))$$

and

$$e_1 = ((r_1, r_2, \dots, r_m, 1), \kappa_1(r_1, r_2, \dots, r_m, 1))$$

for some vertices $(q_1, q_2, \dots, q_m, 0)$ and $(r_1, r_2, \dots, r_m, 1)$ in $G(\langle S_1, S_2, \dots, S_m, 2\rangle)$ and $\kappa_0, \kappa_1 \in \Upsilon$. By reversing the role of the two copies of $G(\langle S_1, \dots, S_m, 1\rangle)$ in $G(\langle S_1, \dots, S_m, 1\rangle) \times e$, we conclude that there is another Hamilton cycle H' in $G(\langle S_1, \dots, S_m, 2\rangle)$ containing edges e'_0 and e'_1 where

$$e'_0 = ((q_1, q_2, \dots, q_m, 1), \kappa_0(q_1, q_2, \dots, q_m, 1))$$

and

$$e'_1 = ((r_1, r_2, \dots, r_m, 0), \kappa_1(r_1, r_2, \dots, r_m, 0)).$$

Let $x = 2t$, and define a parameterized function ψ_i, $0 \le i < t$, that maps the vertices of $G(\langle S_1, \dots, S_m, 2\rangle)$ to the vertices of $G(\langle S_1, \dots, S_m, 2t\rangle)$ by

$$\psi_i((p_1, p_2, \dots, p_m, p_{m+1})) = (p_1, p_2, \dots, p_m, 2i + p_{m+1}), \quad \text{where } p_{m+1} \in \{0, 1\}.$$

Let C_i, $0 \le i < t$, be a cycle of $G(\langle S_1, S_2, \dots, S_m, 2t\rangle)$ that is isomorphic via ψ_i to H if i is even and H' if i is odd. The graph $G(\langle S_1, S_2, \dots, S_m, 2t\rangle)$ is covered by the t cycles $C_0, C_1, \dots, C_{t-1}$; we now show how to join these cycles to construct a Hamilton cycle in $G(\langle S_1, S_2, \dots, S_m, 2t\rangle)$.

To join C_i and C_{i+1} when i is even, note that C_i contains the edge

$$\psi_i(e_1) = ((r_1, r_2, \ldots, r_m, 2i+1), \kappa_1(r_1, r_2, \ldots, r_m, 2i+1))$$

and C_{i+1} contains the edge

$$\psi_{i+1}(\mathrm{e}_1') = ((r_1, r_2, \ldots, r_m, 2i+2), \kappa_1(r_1, r_2, \ldots, r_m, 2i+2)).$$

In addition, $(r_1, r_2, \ldots, r_m, 2i+1)$ is adjacent to $(r_1, r_2, \ldots, r_m, 2i+2)$, and $\kappa_1(r_1, r_2, \ldots, r_m, 2i+1)$ is adjacent to $\kappa_1(r_1, r_2, \ldots, r_m, 2i+2)$ via the τ_{m+1} edges of $G(\langle S_1, S_2, \ldots, S_m, 2t\rangle)$. Delete $\psi_i(e_1)$ and $\psi_{i+1}(e_1')$ and add the edges $((r_1, r_2, \ldots, r_m, 2i+1), (r_1, r_2, \ldots, r_m, 2i+2))$ and $((\kappa_1(r_1, r_2, \ldots, r_m, 2i+1)), \kappa_1(r_1, r_2, \ldots, r_m, 2i+2))$, thus joining C_i and C_{i+1} into a single cycle.

To join C_i and C_{i+1} when i is odd, note that C_i contains the edge

$$\psi_i(\mathrm{e}_0') = ((q_1, q_2, \ldots, q_m, 2i+1), \kappa_0(q_1, q_2, \ldots, q_m, 2i+1))$$

and C_{i+1} contains the edge

$$\psi_{i+1}(e_0) = ((q_1, q_2, \ldots, q_m, 2i+2), \kappa_0(q_1, q_2, \ldots, q_m, 2i+2)).$$

As in the even case, delete $\psi_i(\mathrm{e}_0')$ and $\psi_{i+1}(e_0)$ and add the edges $((q_1, q_2, \ldots, q_m, 2i+1), (q_1, q_2, \ldots, q_m, 2i+2))$ and $(\kappa_0(q_1, q_2, \ldots, q_m, 2i+1), \kappa_0(q_1, q_2, \ldots, q_m, 2i+2))$, thus joining C_i and C_{i+1} into a single cycle.

The cycles $C_0, C_1, \ldots, C_{t-1}$ are now joined to form a Hamilton cycle in $G(\langle S_1, S_2, \ldots, S_m, 2t\rangle)$. $\square$

In this section, we have assumed that the product spaces $\mathbf{S}$ are right-anchored and not frozen; as a result, in the next theorem, the product space $\mathbf{S}' = \langle S_1, S_2, \ldots, S_m\rangle$ has $S_j > 1$ for all $j = 1, 2, \ldots, m$. Hence, $\mathbf{S}'$ is free.

Theorem 4.6. *The graph $G^2(\langle S_1, S_2, \ldots, S_m\rangle)$ is Hamiltonian, if all $S_j > 1$.*

Proof. Let $\mathbf{S}' = \langle S_1, S_2, \ldots, S_m\rangle$. By Theorems 4.3 and 4.1, the graph $G(\langle \mathbf{S}', 1\rangle)$ is isomorphic to the cover graph of a distributive lattice, so by Theorem 4.4, the graph $G(\langle \mathbf{S}', 1\rangle) \times e$ has a Hamilton cycle H. Since, by Lemma 3.7, $G(\langle \mathbf{S}', 1\rangle)$ is bipartite, $G^2(\langle \mathbf{S}', 1\rangle)$ is also Hamiltonian. In fact, if one simply records every other vertex in H, then a Hamilton cycle in $G^2(\langle \mathbf{S}', 1\rangle)$ is obtained. Since $G(\langle \mathbf{S}', 1\rangle)$ is isomorphic to a spanning subgraph of $G(\mathbf{S}')$, $G^2(\mathbf{S}')$ is also Hamiltonian. $\square$

5. Additional properties of the lattice

In this section, we prove some fundamental properties of the lattice that do not have direct bearing on the Hamiltonicity questions considered earlier.

Every distributive lattice is ranked, and the rank of an element $\mathbf{p}$ is $r(\mathbf{p})$, where r was defined in (2). The rank generating function of $P(\mathbf{S})$ is

$$g(\mathbf{S}; z) := \sum_{\mathbf{p}\in\mathbf{S}} z^{r(\mathbf{p})}. \tag{5}$$

Lemma 5.1. *Let $\mathbf{S} = \langle S_1, S_2, \ldots, S_m\rangle$. The rank generating function of $P(\mathbf{S})$ is*

$$g(\mathbf{S}; z) := \sum_{\mathbf{p}\in\mathbf{S}} z^{r(\mathbf{p})} = \prod_{j=1}^{m} \frac{1 - z^{jS_j}}{1 - z^j}.$$

Proof. Let $g_i(z)$ denote $g(\langle S_1, S_2, \ldots, S_i\rangle; z)$. Since $r((p_1, p_2, \ldots, p_{i-1}, j)) = ij + r((p_1, p_2, \ldots, p_{i-1}))$ for $j = 0, 1, \ldots, S_i'$, we have

$$\begin{aligned} g_i(z) &= g_{i-1}(z) + z^i g_{i-1}(z) + \cdots + z^{iS_i'} g_{i-1}(z) \\ &= g_{i-1}(z)(1 + z^i + \cdots + z^{iS_i'}) \\ &= g_{i-1}(z)\frac{1 - z^{iS_i}}{1 - z^i}, \end{aligned}$$

and the result now follows by induction. $\square$

We can use the rank generating function to give an alternate proof of Theorem 3.9. First note that $\mathrm{d}(\mathbf{S}) = g(\mathbf{S}; -1)$ by (5). We can write

$$g(\mathbf{S}; z) = \prod_{j=1}^{m} p_j(z^j), \quad \text{where } p_j(z) = \frac{1 - z^{jS_j}}{1 - z^j} = 1 + z^j + \cdots + z^{jS'_j}.$$

The notation $[\![\Phi]\!]$ for proposition Φ means 1 if Φ is true and 0 if Φ is false. Since $p_j((-1)^j)$ is $[\![S_j \text{ odd}]\!]$ if j is odd, and S_j if j is even,

$$g(\mathbf{S}; -1) = p_1(-1)p_2(+1)\cdots p_m((-1)^m) = \prod_{j \text{ odd}} [\![S_j \text{ odd}]\!] \prod_{j \text{ even}} S_j,$$

which is equivalent to the expression in Theorem 3.9.

Theorem 5.2. *The set of join irreducible elements of $P(\langle S_1, S_2, \ldots, S_m\rangle)$ is*

$$\{(S'_1, S'_2, \ldots, S'_s, 0, 0, \ldots, 0, x, S'_{t+1}, \ldots, S'_m) \mid 0 \le s < t \le m, 0 \le x < S'_t\}.$$

There are $\sum_j jS'_j$ join irreducible elements.

Proof. An element is join irreducible if it is covered by exactly one element. The elements in the set above are clearly join irreducible. If $s = 0$, then only τ_1 can be applied, and if $s > 0$ then only σ_s can be applied. Conversely, if an element is not of that form, then more than one operation (τ_1 or σ_j) can be applied, unless the element is maximum (i.e., $(S'_1, S'_2, \ldots, S'_m)$). The number of elements in the set is

$$\sum_{t=1}^{m} \sum_{s=0}^{t-1} \sum_{x=0}^{S'_j - 1} = \sum_{t=1}^{m} \sum_{s=0}^{t-1} S'_t = \sum_{t=1}^{m} t S'_t.$$

As is always the case, this is the rank of the maximum element. □

Lemma 5.3. *The poset $P(\mathbf{S})$ is self-dual.*

Proof. If $\mathbf{p} = (p_1, p_2, \ldots, p_m)$ is a polyomino, then define its *dual* as

$$\overline{\mathbf{p}} = \overline{(p_1, p_2, \ldots, p_m)} := (S'_1 - p_1, S'_2 - p_2, \ldots, S'_m - p_m).$$

Clearly, $\overline{\overline{\mathbf{p}}} = \mathbf{p}$. It is easy to check that $\mathbf{p} \prec \mathbf{q}$ implies that $\overline{\mathbf{q}} \prec \overline{\mathbf{p}}$. □

There are many other fascinating properties of this lattice, but those will have to await a follow-up paper.

6. Bounding box gray codes

In this last section, we return to Gray codes for the $[a_1, a_2, \ldots, a_{k+1}]$-ominoes that only use τ_j, $1 \le j \le k$, operations. Such Gray code listings are easily produced by the generalized reflected Gray code for mixed-radix numbers [6]. In this case, the closeness graph is a multi-dimensional grid graph.

As previously mentioned, we have two choices at each move depending on whether the left or right part of the polyomino moves during each τ operation. That is, thinking of the polyomino being laid out on the integer grid, a τ_j^+ move is accomplished by either moving the columns $1, 2, \ldots, j$ up one, or by moving the columns $j+1, \ldots, k+1$ down one.

If we are only concerned about generating the different shapes, then this choice makes no difference, but if we are thinking about the polyominoes as being embedded in the plane, then we can consider new problems that take advantage of these choices. For example, for a polyomino screen-saver, we might want the polyominoes to march across the screen as they are being generated. Alternatively, we might want all the polyominoes to remain in more or less the same area on the screen. The authors have implemented a polyomino generator that allows the user to fix some column in the plane and have the move choices made appropriately (i.e., always moving the part of the polyomino that does not contain the fixed column).

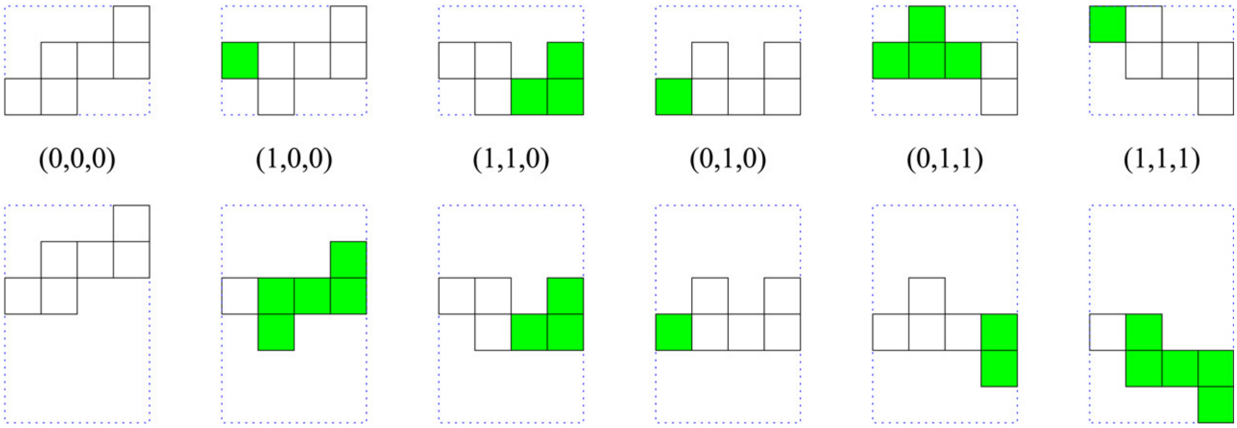


Fig. 6. The same Gray code for [1, 2, 1, 2]-ominoes but with different bounding boxes.

Let us define the *bounding box* of a polyomino Gray code to be the smallest rectangle that contains all the polyominoes as they are successively generated. To illustrate these ideas, we show in Fig. 6 the same Gray code, but with different choices made for which half moves. In the upper listing, the bounding box is as small as possible, and in the lower listing, the bounding box is as large as possible. The shaded columns are the columns that move. It is natural to ask: Which of the many Gray codes for the $[a_1, a_2, \ldots, a_{k+1}]$-ominoes has the smallest (largest) bounding box? Of course, since the width of the box is fixed at $k + 1$, only its height can vary.

Lemma 6.1. *The smallest bounding box of a Gray code for the* $[a_1, a_2, \ldots, a_{k+1}]$*-ominoes has height* $a_1 + a_2 + \cdots + a_{k+1} - k$*.*

Proof. The quantity $H = a_1 + a_2 + \cdots + a_{k+1} - k$ is the height of the polyomino corresponding to $(0, 0, \ldots, 0)$, so the height of the bounding box can be no smaller. Conversely, any Gray code can be forced to remain in a fixed box of height H by simply moving the other half if a move would move some half out of the box. □

Lemma 6.2. *Let* $\mathbf{a} = [a_1, a_2, \ldots, a_{k+1}]$ *and* $\langle S_1, S_2, \ldots, S_k\rangle = \vartheta(\mathbf{a})$*. There is a Gray code for the* **a***-ominoes and a sequence of move choices whose bounding box has height at least* $(S_1 S_2 \cdots S_k)/2$*.*

Proof. Recall that the closeness graph is a k-dimensional grid graph. Suppose that some S_j is even, then there is a Hamilton cycle in the closeness graph. In that cycle, for every j, the number of τ_j^+ edges is the same as the number of τ_j^- edges. Assume that we start at the **0** vertex, so that the first edge is a plus edge. The last edge in the cycle is a minus edge and is removed to get a Hamilton path, H. Adopt the following rule along this path: move the left half (down) on every τ_j^- move. This will result in the left column moving down $(S_1 S_2 \cdots S_k)/2 - 1$ times. The path H is our Gray code.

In the case where all S_j are odd, there is no Hamilton cycle. However, there will be a Hamilton path in which the starting vertex is $(0, 0, \ldots, 0)$ and the ending vertex is $(2, 0, \ldots, 0)$. In this path, the number of plus edges is two greater than the number of minus edges. By the same argument as in the previous case, this will give a Gray code where the left column moves down $(S_1 S_2 \cdots S_k - 1)/2 - 1$ times.

To get the result stated in the lemma we need only observe that the **0** polyomino rises at least two cells above the leftmost column. □

7. Conclusion and open problems

In this paper, we mainly established necessary conditions for the existence of Gray codes for the **a**-ominoes, where only one cell is allowed to move at each step of the Gray code. As far as we know, these conditions could also be sufficient. We are thus led to ask the following questions:

Question 1: Is $G(\langle S_1, S_2, \ldots, S_k\rangle)$ Hamiltonian if k is even and all $S_i \geq 2$ (refer to Lemma 3.8)?

Question 2: Is $G(\langle S_1, S_2, \ldots, S_k\rangle)$ Hamiltonian if k is odd, all $S_i \geq 2$, and there is some odd j such that S_j is even (refer to Corollary 3.10)?

Question 3: Does $G(\langle S_1, S_2, \ldots, S_k, 1\rangle)$ have a Hamilton path whenever $S_1 S_2 \cdots S_k$ and $\sum_{j=1}^{k} j S'_j$ have opposite parities (refer to Lemma 3.11)?

In Theorem 4.6, we showed that if 2 cells are allowed to move at each step, there is a Gray code for the **a**-ominoes when $\vartheta(\mathbf{a})$ is free. However, the 2 cells that move may be in different columns.

Question 4: If $\vartheta(\mathbf{a})$ is free, is there a Gray code for the **a**-ominoes where one or two cells *in the same column* move at each step?

And finally, it would be interesting to extend some of these results/ideas to more general classes of polyominoes. We list one such question below.

Question 5: Is there a Gray code for general column-convex polyominoes where one cell moves at each step? In this case a cell would have to be able to move between columns.

Acknowledgements

We wish to thank the organizers of the BIRS (Banff International Research Station) workshop on Generalized De Bruijn Cycles and Gray Codes, held on December 4–9, 2004. The results of this paper were first presented at that workshop. We thank Scott Craig for making a number of useful suggestions. We are also grateful to the referees for their perceptive comments that improved the paper.

Research of Stirling Chow supported in part by an NSERC Canada Graduate Scholarship. Research of Frank Ruskey supported in part by an NSERC discovery grant.

References

[1] S. Gill Williamson, Combinatorics for Computer Science, Computer Science Press, 1985.
[2] Solomon Golomb, Polyominoes: Puzzles, Patterns, Problems, and Packings, 2nd ed., Princeton University Press, 1994.
[3] Dean Hickerson, Counting horizontally convex polyominoes, Journal of Integer Sequences 2 (1999). http://www.math.uwaterloo.ca/JIS/HICK2/chcp.html.
[4] Iwan Jensen, Anthony J. Guttmann, Statistics of lattice animals (polyominoes) and polygons, Journal of Physics, Ser. A 33 (2000) L257–L263.
[5] David A. Klarner, Some results concerning polyominoes, Fibonacci Quarterly 3 (1965) 9–20.
[6] Donald E. Knuth, The Art of Computer Programming, volume 4, Fascicle 2: Generating All Tuples and Permutations, Addison-Wesley, 2005.
[7] S. Mertens, Counting lattice animals: a parallel attack, Journal of Statistical Physics 66 (1992) 669–678.
[8] Gara Pruesse, Frank Ruskey, Gray codes from antimatroids, Order 10 (1993) 239–252.
[9] D.H. Redelmeier, Counting polyominoes: yet another attack, Discrete Mathematics 36 (1981) 191–203.
[10] Carla D. Savage, Ian Shields, Douglas B. West, On the existence of Hamiltonian paths in the cover graph of $M(n)$, Discrete Mathematics 262 (2003) 241–252.
[11] Neil J.A. Sloane, The On-Line Encyclopedia of Integer Sequences. http://www.research.att.com/~njas/sequences/.
[12] Richard Stanley, Enumerative Combinatorics, Cambridge University Press, 2000.